




Handwritten signature/initials

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	§	
Christina Hsu et al.	§	Group Art Unit: 2192
	§	
Serial No.: 10/677,004	§	Confirmation No.: 7237
	§	
Filed: October 1, 2003	§	Examiner: Dao, Thuy Chan
	§	
For: Method and Apparatus for	§	Atty. Docket: 200208014-1
Supporting Configuration of A Web	§	NUHP:0121/FLE/LIU
Application in a Web Presentation	§	
Architecture	§	

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF TRANSMISSION OR MAILING 37 C.F.R. 1.8
I hereby certify that this correspondence is being transmitted by facsimile to the United States Patent and Trademark Office in accordance with 37 C.F.R. § 1.6(d), or is being transmitted via the Office electronic filing system in accordance with 37 C.F.R. § 1.6(a)(4), or is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date below.
October 6, 2008 
Date Katey Hines

**RESPONSE TO NOTIFICATION OF NON-COMPLIANT
APPEAL BRIEF PURSUANT TO 37 C.F.R. §§ 41.31 AND 41.37**

In response to the Notification of Non-Compliant Appeal Brief mailed on September 24, 2008, Appellants resubmit the previous Appeal Brief (filed September 2, 2008), revised to address the above-referenced notification of non-compliance. In particular, Appellants note that Section 5 (Summary of the Claimed Subject Matter) is amended in accordance with the Examiner's remarks in the Notification to include a summary of independent claim 22. It is respectfully submitted that these revisions are sufficient to place the Appeal Brief in compliance with 37 C.F.R. §§ 41.31 and 41.37.

This Appeal Brief is being filed in furtherance to the Notice of Appeal mailed on June 27, 2008, and received by the Patent Office on July 2, 2008. Because the requisite fee of \$510.00 was previously paid, Appellants do not believe any fees are due at this time. However, if any fees are due, the Commissioner is authorized to charge any fees that may be necessary to advance prosecution of the present application, to Account No. 08-2025, Order No. 200208014-1 (NUHP:0121/FLE).

1. **REAL PARTY IN INTEREST**

The real party in interest is Hewlett-Packard Development Company, L.P., the Assignee of the above-referenced application by virtue of the Assignment to Hewlett-Packard Development Company, LP, recorded at reel 014581, frame 0608, on October 1, 2003. Accordingly, Hewlett-Packard Development Company, L.P., will be directly affected by the Board's decision in the pending appeal.

2. **RELATED APPEALS AND INTERFERENCES**

Appellants are unaware of any other appeals or interferences related to this Appeal. The undersigned is Appellants' legal representative in this Appeal.

3. **STATUS OF CLAIMS**

Claims 1-24 are currently pending, are currently under final rejection and, thus, are the subject of this Appeal.

4. **STATUS OF AMENDMENTS**

In the Advisory Action mailed on June 19, 2008 ("Advisory Action"), the Examiner stated that certain amendments were entered for the purposes of the present Appeal. *See* Advisory Action, page 1. This statement appears to be erroneous, as no claim amendments were submitted in response to the Final Office Action mailed March 28, 2008 ("Final Office Action"). *See* Final Office Action, pages 2-6. Accordingly, Appellants believe there are no outstanding amendments to be considered by the Board.

5. **SUMMARY OF CLAIMED SUBJECT MATTER**

The present Application contains four independent claims, namely, claims 1, 8, 15 and 22, as well as dependent claims 2, 3, 9, 10, 16, and 17. The subject matter of these claims, which is the subject of the present Appeal, is summarized below.

Claims 1-3, 8-10, and 15-17 generally relate to systems and techniques involved in providing a web presentation architecture (WPA) for the creation, support, and configuration of web-based applications. A WPA, such as the WPA 100 generally illustrated by Fig. 2, may be adapted to execute on a processor-based device, such as a computer system or the like, and may include various components. The WPA 100, in certain embodiments, may include a configurator that functions to load configuration information required by one or more web-based applications into system memory at startup. *See, e.g.*, Application, paragraph 30. The loaded configuration information may be retained in system memory for the duration of an application server session and be used by subsequent HTTP requests received by the server, thereby improving the performance of web-based applications by eliminating the need to reload the information each time the server receives a request. *See id.* at paragraph 34.

With regard to the aspect of the invention set forth in independent claim 1, discussions of the recited features of claim 1 can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in accordance with claim 1 provides a system (*e.g.*, 100) for creating web applications (*e.g.*, 204). The system includes a controller generator (*e.g.*, 102) that is adapted to provide a

web application (*e.g.*, 204) with a controller (*e.g.*, 18, 208) that receives a request (*e.g.*, 148) for data from a user (*e.g.*, represented by client 14) and responds to the request by sending information (*e.g.*, 150) to the user (*e.g.*, 14). *See, e.g., id.* at paragraphs 10, 15, 18, 19-22, 24, 36; *see also* Figs. 1-3. The system further includes a configurator generator (*e.g.*, noting the configuration service manager 116) that is adapted to provide a configurator (*e.g.*, 210) that loads configuration information (*e.g.*, stored in cached configuration file 212) for use by the controller (*e.g.*, 18, 208) from a cached configuration file (*e.g.*, 212) that originated from a backend data store (*e.g.*, databases or service based frameworks, 138, 140), wherein the configurator (*e.g.*, 210) stores the configuration information for subsequent access (*e.g.*, in the object cache manager 114). *See, e.g., id.* at paragraphs 29-30, 34-41; *see also* Figs. 2-3.

Next, with regard to the aspect of the invention set forth in dependent claim 2, discussions of the recited features of claim 2 can be found at least in the locations in the specification and drawings cited below. By way of example, an embodiment in accordance with claim 2 relates to the system of claim 1 (see above), wherein the configuration file (*e.g.*, 212) is a text properties configuration file. *See, e.g., id.* at paragraphs 13-14, 16, 37; *see also* Figs. 1, 3.

Next, with regard to the aspect of the invention set forth in dependent claim 3, discussions of the recited features of claim 3 can be found at least in the locations in the specification and drawings cited below. By way of example, an embodiment in accordance with claim 3 relates to the system of claim 1 (see above), wherein the

configurator (*e.g.*, 210) is adapted to store the configuration information (*e.g.*, 212) as a singleton object (*e.g.*, 214). *See, e.g., id.* at paragraphs 34, 37-39; *see also* Fig. 3.

Next, with regard to the aspect of the invention set forth in independent claim 8, discussions of the recited features of claim 8 can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in accordance with claim 8 provides a method of creating web applications (*e.g.*, 204). The method includes creating, with a processor-based device, a controller (*e.g.*, 18, 208) that receives a request (*e.g.*, 148) for data from a user (*e.g.*, represented by client 14) and responds to the request by sending information (*e.g.*, 150) to the user (*e.g.*, 14). *See, e.g., id.* at paragraphs 10, 15, 18, 19-22, 24, 36; *see also* Figs. 1-3. The method further includes providing a configurator (*e.g.*, 210) that loads configuration information (*e.g.*, stored in cached configuration file 212) for use by the controller (*e.g.*, 18, 208) from a cached configuration file (*e.g.*, 212) that originated from a backend data store (*e.g.*, databases or service based frameworks, 138, 140), and wherein the configurator (*e.g.*, 210) stores the configuration information for subsequent access. *See, e.g., id.* at paragraphs 29-30, 34-41; *see also* Figs. 2-3.

Next, with regard to the aspect of the invention set forth in dependent claim 9, discussions of the recited features of claim 9 can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in accordance with claim 9 relates to the method of claim 8 (see above), wherein the

configuration file (*e.g.*, 212) is defined to be a text properties configuration file. *See, e.g., id.* at paragraphs 13-14, 16, 37; *see also* Figs. 1, 3.

Next, with regard to the aspect of the invention set forth in dependent claim 10, discussions of the recited features of claim 10 can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in accordance with claim 10 relates to the method of claim 8 (see above), wherein the step of providing the configurator (*e.g.*, 210) includes providing a configurator (*e.g.*, 210) adapted to store configuration information as a singleton object (*e.g.*, 214). *See, e.g., id.* at paragraphs 34, 37-39; *see also* Fig. 3.

Next, with regard to the aspect of the invention set forth in independent claim 15, discussions of the recited features of claim 15 can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in accordance with claim 15 provides a system (*e.g.*, 100) for creating web applications (*e.g.*, 204). The system includes means (*e.g.*, processor-based device and/or computer system executing a web presentation architecture 100 to create the elements Model-View-Controller 10 – model 12, view 16, and *controller* 18) for creating a controller (*e.g.*, 18, 208) that is adapted to receive a request (*e.g.*, 148) for data from a user (*e.g.*, 14) and respond to the request (*e.g.*, 150). *See, e.g., id.* at paragraphs 10, 11, 15, 18, 19-22, 24, 36; *see also* Figs. 1-3. The system further includes means (*e.g.*, configuration manager architecture 116 constructs one or more configurators) for creating a configurator (*e.g.*, 210) that loads configuration information (*e.g.*, stored in cached

configuration file 212) for use by the controller (*e.g.*, 18, 208) from a cached configuration file (*e.g.*, 212) that originated from a backend data store (*e.g.*, databases or service based frameworks, 138, 140), and wherein the configurator (*e.g.*, 210) stores the configuration information for subsequent access. *See, e.g., id.* at paragraphs 29-30, 34-41; *see also* Figs. 2-3.

Next, with regard to the aspect of the invention set forth in dependent claim 16, discussions of the recited features of claim 16 can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in accordance with claim 16 relates to the system of claim 15 (see above), wherein the configuration file (*e.g.*, 212) is a text properties configuration file. *See, e.g., id.* at paragraphs 13-14, 16, 37; *see also* Figs. 1, 3.

Next, with regard to the aspect of the invention set forth in dependent claim 17, discussions of the recited features of claim 17 can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in accordance with claim 17 relates to the system of claim 15 (see above), wherein the configurator (*e.g.*, 210) is adapted to store the configuration information (*e.g.*, 212) as a singleton object (*e.g.*, 214). *See, e.g., id.* at paragraphs 34, 37-39; *see also* Fig. 3.

Finally, with regard to the aspect to the invention set forth in independent claim 22, discussions of the recited features of claim 22 can be found at least in the below-cited locations of the specification and drawings. By way of example, an embodiment in

accordance with claim 22 relates to a machine readable medium. The machine readable medium (*e.g.*, the web server 202 hosts, and thus *stores* on one or more machine readable media, the web application 204 which may include a controller 208 and a configurator 210) includes a controller generator (*e.g.*, 102) stored thereon, the controller generator (*e.g.*, 102) being adapted to provide a web application (*e.g.*, 204) with a controller (*e.g.*, 18, 208) that receives a request (*e.g.*, 148) for data from a user (*e.g.*, represented by client 14) and responds to the request by sending information (*e.g.*, 150) to the user. *See, e.g., id.* at paragraphs 10, 11, 15, 18, 19-22, 24, 36; *see also* Figs. 1-3. The machine readable medium further includes a configurator generator (*e.g.*, noting the configuration service manager 116) stored thereon, the configurator generator being adapted to provide a configurator (*e.g.*, 210) that loads configuration information (*e.g.*, stored in cached configuration file 212) for use by the controller (*e.g.*, 18, 208) from a cached configuration file (*e.g.*, 212) that originated from a backend data store (*e.g.*, databases or service based frameworks, 138, 140), and wherein the configurator (*e.g.*, 210) stores the configuration information for subsequent access (*e.g.*, in the object cache manager 114). *See, e.g., id.* at paragraphs 29-30, 34-41; *see also* Figs. 2-3.

6. **GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

First Ground of Rejection for Review on Appeal:

Appellants respectfully urge the Board to review and reverse the Examiner's first ground of rejection in which the Examiner rejected claims 1-7 and 15-21 under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

Second Ground of Rejection for Review on Appeal:

Appellants respectfully urge the Board to review and reverse the Examiner's second ground of rejection in which the Examiner rejected claims 1-24 under 35 U.S.C. § 102(b) as anticipated by Hutsch et al., U.S. Patent Application Publication No. 2001/0034771 A1 (hereinafter "the Hutsch reference").

7. **ARGUMENT**

As discussed in detail below, the Examiner has improperly rejected the pending claims. Further, the Examiner has misapplied long-standing and binding legal precedents and principles in rejecting the claims under Sections 101 and 102. Accordingly, Appellants respectfully request that the Board to reverse all the pending rejections.

As an initial matter, Appellants note that the Examiner objected to independent claims 1, 8, 15, and 22 in the Final Office Action. *See* Final Office Action, pages 2-3. In particular, the Examiner alleged that the specification fails to provide support for the caching of configuration files, as well as the storing of cached configuration information, features which are commonly recited by each of claims 1, 8, 15, and 22. In the previously filed Response to the Final Office Action, Appellants provided a detailed explanation and cited to various portions of the specification which are believed to support these claimed features. *See* Response to Final Office Action, pages 7-10.

In the Advisory Action, the Examiner stated that Appellants' explanation and provided support are not persuasive to overcome the objection. *See* Advisory Action,

page 2. Although Appellants strongly disagree with the Examiner's position and further believe that the claimed subject matter is fully supported by the specification, it should be noted that the Examiner has never issued a formal rejection with regard to these claimed features. Therefore, in view of the Examiner's failure to formally reject independent claims 1, 8, 15, and 22 for these reasons, Appellants submit that this issue with regard to the objection of claims 1, 8, 15, and 22 is not ripe for consideration before the Board. However, Appellants reserve the right to provide further comment in a reply brief if necessary.

A. **First Ground of Rejection:**

With respect to the Examiner's rejection of claims 1-7 and 15-21 under 35 U.S.C. § 101 as being directed to non-statutory subject matter, the Examiner stated the following:

As set forth in the previous Office action mailed September 12, 2007, January 26, 2007 and August 7, 2006, claims 1-7 and 15-21 are rejected because the claimed invention is directed to non-statutory subject matter. They amount to Functional Descriptive Material: "Data Structures" representing descriptive material per se or "Computer Programs" representing computer listings per se.

Claims 1 and 15

Claims 1 and 15 recite "A system for creating web applications ...", which may comprise only software components (i.e., "*a controller generator*" 102 and "*a configurator generator*" 116, which can be implemented as a Servlet, [0019], lines 7-10).

Data structures not claimed as embodied in computer-readable media are descriptive material per se and are not

statutory because they are not capable of causing functional change in the computer. See, e.g., Warmerdam, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure per se held nonstatutory). Such claimed data structures do not define any structural and functional interrelationships between the data structure and other claimed aspects of the invention which permit the data structure's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory (emphasis added).

Similarly, computer programs claimed as computer listings per se, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. See Lowry, 32 F.3d at 1583-84, 32 USPQ2d at 1035. Accordingly, it is important to distinguish claims that define descriptive material per se from claims that define statutory inventions (emphasis added). See MPEP 2106.01 (I).

Claims 2-7 and 16-21:

Claims 2-7 and 15-21 further recite functional descriptions of said software components and do not remedy the deficiencies of independent claims 1 and 15, respectively.

Final Office Action, pages 3-5. (Emphasis in original).

Appellants respectfully submit to the Board that the Examiner's rejection of claims 1-7 and 15-21 under 35 U.S.C. § 101 is clearly erroneous and that these claims are directed towards statutory subject matter. In particular, Appellants direct the Board's attention to the language set forth under 35 U.S.C. § 101, which states:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title.

35 U.S.C. § 101.

In interpreting these provisions of Section 101, the Supreme Court stated that Congress intended statutory subject matter to "include *anything* under the sun that is made by man." *Diamond v. Chakrabarty*, 447 U.S. 303, 309, 206 U.S.P.Q. 193, 197 (1980) (emphasis added). Although this statement may appear limitless, the Supreme Court has identified three categories of unpatentable subject matter: laws of nature, natural phenomena, and abstract ideas. *See Diamond v. Diehr*, 450 U.S. 175, 182, 209 U.S.P.Q. 1, 7 (1981). Accordingly, so long as a claim is not directed to one of the three specific areas listed above, the claim is directed to patentable subject matter. Thus, it is improper for the Examiner to read restrictions into Section 101 regarding subject matter that may be patented where the legislative history does not indicate that Congress clearly intended such a limitation. *In re Alappat*, 31 U.S.P.Q.2d 1545, 1556 (Fed. Cir. 1994) (citing *Chakrabarty* 447 U.S. at 308).

Moreover, the Federal Circuit has held that the mere fact that a claim includes or is directed to an algorithm is no ground for holding that a claim is directed to non-statutory subject matter. *See In re Iwashashi*, 12 U.S.P.Q.2d 1908, 1911 (Fed. Cir 1989). Rather, the proscription against patenting an algorithm, to the extent it still exists, is narrowly limited to *mathematical algorithms in the abstract* (e.g., describing a mathematical algorithm as a procedure for solving a given type of mathematical problem). *See AT&T Corp. v. Excel Communications, Inc.*, 50 U.S.P.Q.2d 1447, 1450 (Fed. Cir 1999). Indeed, the courts are well aware that any patentable step-by-step process, be it electronic, chemical, or mechanical, may innately involve an algorithm. *Id.* at 1450.

With the above points in mind, Appellants submit that an inquiry into what qualifies as statutory subject matter simply requires “an examination of the contested claims to see if the claimed subject matter as a whole is a disembodied mathematical concept representing nothing more than a ‘law of nature’ or an ‘abstract idea, or if the mathematical concept has been reduced to some practical application rendering it ‘useful.’” *Id.* at 1451 (citing and quoting *In re Alappat*, 31 U.S.P.Q.2d at 1557). Furthermore, a Section 101 analysis “demands that the focus in any statutory subject matter analysis be on the *claim as a whole*.” *In re Alappat*, 31 U.S.P.Q.2d at 1557 (citing *Diehr*, 450 U.S. at 192) (emphasis in original). Indeed, the dispositive inquiry is whether the claim *as a whole* is directed to statutory subject matter; it is irrelevant that a claim may contain, as part of the whole, subject matter that would not be patentable by itself. *Id.*

Independent Claims 1 and 15 are Directed to Statutory Subject Matter

Appellants respectfully disagree with the Examiner's assertions and interpretation of the law under 35 U.S.C. § 101. In particular, Appellants note that the Examiner's rejection appears to be based on the mistaken assumption that the subject matter recited by independent claims 1 and 15 are purely software (e.g., alleged by the Examiner as "purely Data Structures"). *See* Final Office Action, pages 3-5. As will be explained in detail below, this assumption is clearly erroneous.

Independent claims 1 and 15 each recite "a *system* for creating web applications." (Emphasis added). Independent claim 1 further recites, *inter alia*, "a controller generator ... adapted to provide a web application with a controller that *receives* a request for data ... and *sending* information to the user" and "a configurator generator ... adapted to provide a configurator that *loads* configuration information." (Emphasis added). Independent claim 15 similarly recites, *inter alia*, "means for creating a controller ... adapted to *receive* a request for data" and "means for creating a configurator that *loads* configuration information." (Emphasis added).

First, Appellants respectfully submit to the Board that the recited "configurator generator" and "controller generator" of independent claim 1 would clearly be understood by one skilled in the art to encompass more than "only software components," as suggested by the Examiner. *See id.* at page 4. For instance, the "configurator generator" is recited as performing the function of providing a configurator for *loading*

configuration information. Similarly, the “controller generator” is recited as providing a controller for *receiving* and *sending* information. Thus, each of these recited elements clearly carries out specific *functions* which could not be accomplished by software alone. That is, software, by itself, is simply a listing of instructions, encoded or otherwise. While such instructions may indeed define one or more specific functions, it is generally well understood that *at least some hardware* is required to execute the instructions defined by software in order to carry out or perform the functions defined therein. Indeed, Appellants submit that one skilled in the art would clearly understand that the recited controller generator and configurator generator each includes at least some hardware components for carrying out the functions of *loading*, *storing*, and *receiving* information. Therefore, contrary to the Examiner’s view, Appellants respectfully submit to the Board that each of the elements recited by independent claims 1 and 15 must encompass *more* than just software. As such, each of independent claims 1 and 15 are believed to be statutory under Section 101.

As an ancillary note, Appellants further submit to the Board that the preamble of each of independent claims 1 and 15, which recites “a system,” and would be readily appreciated by one skilled in the art as encompassing *at least some hardware elements*. Moreover, referring to independent claim 22, Appellants note that claim 22 recites application instructions (i.e., software) stored on a *machine readable medium* for performing the same functionality generally recited in independent claims 1 and 15. Thus, the doctrine of claim differentiation compels an interpretation that the recited systems of claims 1 and 15 include at least some hardware elements and, thus, cannot

simply be interpreted as software *per se*. Indeed, as noted by Appellants in the previously filed Response to the Final Office Action, *Webster's Ninth New Collegiate Dictionary* defines the term "system," when used in the present context, as being a "group of devices or artificial objects or an organization forming a network esp. for distributing something or serving a common purpose." See Response to Final Office Action mailed March 28, 2007, page 16; *see also Webster's Ninth New Collegiate Dictionary*, page 1199 (1989). A copy of this dictionary definition is provided as Exhibit A in the Appendix of Evidence attached hereto.

Additionally, the present application *clearly* describes embodiments of the disclosed web presentation architecture (WPA) as being "adapted to execute on a processor-based device such as a computer system or the like." Application, paragraph 17. (Emphasis added). Further, the system recited by claim 15 includes "means for" language, which should be addressed in accordance with M.P.E.P. § 2181. Accordingly, Appellants believe that the system recited in claim 15 includes sufficient structure (e.g., processor-based device) to comply with the guidelines set forth under M.P.E.P. § 2106.01.

With the foregoing discussion in mind, Appellants respectfully note that the M.P.E.P. makes it clear that a claim directed towards a "statutory manufacture or machine," such as a processor-based device or computer system, is statutory *even if* certain elements (e.g., the configurator generator and controller generator) may include

software. In particular, Appellants direct the Board's attention to Section 2106.01(I) of the M.P.E.P. which states, in relevant part:

Computer programs are often recited as part of a claim. USPTO personnel should determine whether the computer program is being claimed as part of an otherwise statutory manufacture or machine. In such a case, *the claim remains statutory irrespective of the fact that a computer program is included in the claim.* The same result occurs when a computer program is used in a computerized process where the computer executes the instructions set forth in the computer program. Only when the claimed invention taken as a whole is directed to a mere program listing, i.e., to only its description or expression, is it descriptive material *per se* and hence nonstatutory.

Since a computer program is merely a set of instructions capable of being executed by a computer, the computer program itself is not a process and USPTO personnel should treat a claim for a computer program, without the computer-readable medium needed to realize the computer program's functionality, as nonstatutory functional descriptive material. When a computer program is claimed in a process where the computer is executing the computer program's instructions, USPTO personnel should treat the claim as a process claim. *When a computer program is recited in conjunction with a physical structure, such as a computer memory, USPTO personnel should treat the claim as a product claim.*

M.P.E.P. § 2106.01(I). (Emphasis added).

Therefore, even though certain elements recited by independent claims 1 and 15 may include software, Appellants submit that claims 1 and 15 are *clearly* statutory under Section 101 because the software elements are being claimed as *part* of a "system." The

claims are *not*, as the Examiner incorrectly suggests, directed towards pure data structures or program listings.

Further, Appellants note that the Examiner's apparent reliance on *In re Warmerdam* in setting forth the present rejection is misplaced. See Final Office Action, page 4. In *Warmerdam*, the Federal Circuit concluded that the applicant's method claims were directed to nonstatutory subject matter because they related to *method steps* involving the mere manipulation of *abstract ideas*. However, with respect to the applicant's claim directed to "a machine," the court stated that those claims were "clearly patentable subject matter." *Warmerdam*, 31 U.S.P.Q. 2d at 1759. (Emphasis added). With this distinction in mind, Appellants submit that the "system" recited by each of independent claims 1 and 15 is *clearly* directed towards a machine (e.g., processor-based device and/or computer system) and, therefore, believed to be clearly statutory in view of *Warmerdam*.

With regard to the Advisory Action, Appellants note that the Examiner has provided no additional support for the Section 101 rejections of claims 1-7 and 15-21 other than merely restating the erroneous positions previously set forth in the Final Office Action and addressed above. See Advisory Action, page 2. Thus, for at least the reasons presented above, Appellants contend that the rejections of independent claims 1 and 15 under 35 U.S.C. § 101 are improper. Accordingly, Appellants respectfully request that the Board overturn the Examiner's rejection of claims 1-7 and 15-21 under 35 U.S.C. § 101 and instruct the Examiner to allow these claims.

B. Second Ground of Rejection:

The Examiner further rejected claims 1-24 under 35 U.S.C. § 102(b) as being anticipated by the Hutsch reference. Specifically, with regard to independent claims 1, 8, 15, and 22, the Examiner stated in relevant part:

Claim 1:

Hutsch discloses *a system for creating web applications* (e.g., FIG. 3A, [0115]; FIG. 8, [0234-0245]), *the system comprising:*

a controller generator that is adapted to provide a web application with a controller that receives a request for data from a user and responds to the request by sending information to the user (e.g., FIG. 8, Web Server 320 provides web applications to Client Browser 304 after receiving HTTP request, [0237-0245]; [0178-0181]); *and*

a configurator generator that is adapted to provide a configurator that loads configuration information for use by the controller from a cached configuration file (e.g., [0239], FIG. 8, Configuration Service 336 having configuration information, [0156]; configuration information stored in user/application profiles, [0310]-[0318]; user/application profiles as XML files, [0321]-[0323]);

that originated from a backend data store (e.g., FIG. 15, Configuration Back End Databases 337, [0326]; Cache 1560 for data originated from the Configuration Back End Databases 337, [0346-0356]); FIG. 8, block 336), *and*

wherein the configurator stores the configuration information for subsequent access (e.g., FIG. 8, storing the configuration information from user/application profiles in Configuration Service 336 to Profiling Service Configuration File 802 for subsequent access, [0239]).

Claim 8:

Hutsch discloses *a method of creating web applications, the method comprising:*

creating, with a processor-based device, a controller that receives a request for data from a user and responds to the request by sending information to the user (e.g., FIG. 8, Web Server 320 provides web applications to Client Browser 304 after receiving HTTP request, [0237-0245]; [0178-0181]); and

providing a configurator that loads configuration information for use by the controller from a cached configuration file and wherein the configurator stores the configuration information for subsequent access (e.g., [0239], Configuration Service 336 having configuration information, [0156]; configuration information stored in user/application profiles, [0310]-[0318]; user/application profiles as XML files, [0321]-[0323]);

the cached configuration file that originated from a backend data store (e.g., FIG. 15, Configuration Back End Databases 337, [0326]; Cache 1560 for data originated from the Configuration Back End Databases 337, [0346-0356]); caching configuration information for subsequent access, [0327-0329]).

Claim 15:

Hutsch discloses *a system for creating web applications, the system comprising:*

means for creating a controller that is adapted to receive a request for data from a user and respond to the request (e.g., FIG. 8, Web Server 320 provides web applications to Client Browser 304 after receiving HTTP request, [0237-0245]; [0178-0181]); and

means for creating a configurator that loads configuration information for use by the controller from a cached configuration file and wherein the configurator stores the configuration information for subsequent access (e.g., [0239]; FIG. 8, Configuration Service 336 having

configuration information, [0156]; configuration information stored in user/application profiles, [0310]-[0318]; user/application profiles as XML files, [0321]-[0323]; [0239])

the cached configuration file that originated from a backend data store (e.g., FIG. 15, Configuration Back End Databases 337, [0326]; Cache 1560 for data originated from the Configuration Back End Databases 337, [0346-0356]); caching configuration information for subsequent access, [0239], [0327-0329]).

Claim 22:

Hutsch discloses *a machine readable medium, comprising:*

a controller generator stored on the machine readable medium, the controller generator being adapted to provide a web application with a controller that receives a request for data from a user and responds to the request by sending information to the user (e.g., FIG. 8, Web Server 320 provides web applications to Client Browser 304 after receiving HTTP request, [0237-0245]; [0178-0181]); and

a configurator generator stored on the machine readable medium, the configurator generator being adapted to provide a configurator that loads configuration information for use by the controller from a cached configuration file and wherein the configurator stores the configuration information for subsequent access (e.g., [0239], FIG. 8, Configuration Service 336 having configuration information, [0156]; configuration information stored in user/application profiles, [0310]-[0318]; user/application profiles as XML files, [0321]-[0323]; [0239]);

the cached configuration file that originated from a backend data store (e.g., FIG. 15, Configuration Back End Databases 337, [0326]; Cache 1560 for data originated from the Configuration Back End Databases 337, [0346-0356]); caching configuration information for subsequent access, [0239], [0327-0329]).

Final Office Action, pages 5-8, 10. (Emphasis in original).

Appellants respectfully traverse this rejection. Anticipation under Section 102 can be found only if a single reference shows exactly what is claimed. *See Titanium Metals Corp. v. Banner*, 227 U.S.P.Q. 773 (Fed. Cir.1985). For a prior art reference to anticipate under Section 102, every element of the claimed invention must be identically shown in a single reference. *See In re Bond*, 15 U.S.P.Q.2d 1566 (Fed. Cir.1990). That is, the prior art reference must show the *identical invention “in as complete detail as contained in the ... claim”* to support a *prima facie* case of anticipation. *Richardson v. Suzuki Motor Co.*, 9 U.S.P.Q. 2d 1913, 1920 (Fed. Cir. 1989) (emphasis added). Thus, for anticipation, the cited reference must not only disclose all of the recited features but must also disclose the *part-to-part relationships* between these features. *See Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick*, 221 U.S.P.Q. 481, 486 (Fed. Cir.1984). Accordingly, Appellants need only point to a single element or claimed relationship not found in the cited reference to demonstrate that the cited reference fails to anticipate the claimed subject matter.

Features Omitted from Independent Claims 1, 8, 15, and 22

Independent claims 1, 8, 15, and 22, each recite, *inter alia*, “a configurator that loads configuration information for use by the controller from a cached configuration file that originated from a backend store.” (Emphasis added). For example, the present application provides a configuration manager 116 that functions to *oversee the loading of*

frequently used information into memory during the startup of a particular web application. *See* Application, paragraph 30. The frequently used information may include one or more properties or configuration files, which may be cached in memory as a mapper or singleton object. *See id.* Once the required configuration files are loaded into a mapper object, one or more configurators “may be implemented and each may be responsible *for loading configuration information* related to a specific functionality.” *Id.* at paragraph 36.

The Hutsch reference, to the contrary, does not appear to teach or suggest a configuration manager that provides a configurator for *loading configuration information for use by the controller* from a cached configuration file that originated from a backend store as generally recited by each of claims 1, 8, 15, and 22. In setting forth the present rejection, the Examiner appears to have cited the configuration service 336 and the web server 320 of the Hutsch reference as being analogous to the recited configuration manager and controller, respectively. *See* Final Office Action, pages 5-6. However, even assuming such an analogy is proper, Appellants are unable to locate where in the Hutsch reference it is disclosed that the configuration service 336 carries out the *loading of configuration information for use by the controller* (e.g., web server 320) *from a cached configuration file that originated from a backend store.*

In sharp contrast, the Hutsch reference appears to merely disclose that the configuration server 336 *creates* a configuration file, which is then cached internally with respect to the configuration server 336. *See* Hutsch, Fig. 15 (noting, in particular, cache

1560 and DOM Tree 1570). For instance, the Hutsch reference discloses that once initialized, the configuration server 336 creates a DOM tree 1570 which is populated with system policies from a backend database 337. *See id.* at paragraph 346. However, even assuming that the “system policies” from the backend database could be properly construed as “configuration files,” the loading of the configuration data stored in the cached DOM Tree 1570 for use by the web server 320 appears to be performed by *external components*. It is *not*, as the Examiner suggests, loaded by the configuration service 336 itself.

To the contrary, the Hutsch reference is clear that when a client request is received by the web server 320, the task of *loading data* requested by the client is delegated to an appropriate universal content provider 331 selected by a universal content broker 113. *See id.* at paragraph 145; Fig. 3A. By way of example, when a client request includes a request for *configuration information* or data, a component of the universal content broker, such a configuration proxy 1510 associated with a particular content provider 335, accesses the configuration server to *perform the loading of the necessary configuration data* in response to a particular client request received by the web server 320. *See id.* at paragraph 326. Specifically, the configuration proxy 1510 loads or “*gets data from configuration server 336 on behalf of its clients, and caches the data in a DOM tree in proxy 1510.*” *Id.* at paragraph 327. (Emphasis added). Once the configuration data from the configuration server 336 is *loaded by the configuration proxy*, the data may be accessed by clients, applications, and components of the web server 320. *See id.*

In view of the foregoing discussion, Appellants submit to the Board that the Examiner has failed to demonstrate that the Hutsch reference discloses that the configuration server 336 is configured to initiate or perform the *loading of configuration data* (e.g., stored in the DOM tree 1570) for use by a controller (e.g., the web server 320). Instead, the web server 320, as discussed above, relies on one or more configuration proxies 1510 associated with specific content providers 331 for the loading of configuration data which, as *clearly* illustrated in Figs. 3 and 15A of the Hutsch reference, are *separate components* from the configuration service 336. *See id.* at Figs. 3, 15A. As such, Appellants submit that no reasonable analysis of the Hutsch reference could yield an interpretation that the configuration server 336, which the Examiner has asserted as being analogous to the recited configuration generator, performs the function of loading configuration data for use by a controller, as recited by each of independent claims 1, 8, 15, and 22.

Additionally, Appellants note that the Examiner provided no additional analysis or support for the rejection in the Advisory Action other than merely reproducing statements previously set forth in the Final Office Action. Thus, for at least the reasons set forth above, Appellants submit that the Hutsch reference cannot anticipate independent claims 1, 8, 15, and 22, as well as those claims depending therefrom. Accordingly, Appellants respectfully request that the Board overturn the Examiner's rejection of claims 1-24 under 35 U.S.C. § 102(b) and instruct the Examiner to allow these claims.

Features Omitted from Dependent Claims 2, 9, and 16

Claims 2, 9, and 16 depend from independent claims 1, 8, and 15, respectively, and each generally recite that the configuration file originating from the backend store may be a “text properties configuration file.” As described in the present application, a text properties configuration file may generally refer to a configuration file which governs the extrinsic appearance of data when it is output or viewed by a user. *See* Application, paragraph 13. For instance, a text properties configuration file may govern the extrinsic formats of display of data, which may be irrelevant to the intrinsic characteristics of the data (e.g., the actual content or value of the data itself). *See id.* After careful review, Appellants respectfully assert that the Hutsch reference fails to disclose this recited feature.

In rejecting claims 2, 9, and 16 in the Final Office Action, the Examiner relied solely on the following passage of the Hutsch reference:

In still yet another embodiment, the presentation and logic system includes a profiling service and a profiling service configuration file coupled to the profiling service. The profiling service configuration file includes a decision tree wherein the decision tree performs actions. In one embodiment, decision tree is a XML decision tree. The actions performed by the decision tree include: an action based upon request parameters; an action based upon request header parameters; an action based upon user device properties; and an action based upon resource properties.

Hutsch, paragraph 29.

The above passage appears to describe a profiling service configuration file which may include decision logic for processing a plurality of actions based upon request parameters, device properties, or resource properties. However, Appellants are unable to identify *any* teaching in the cited paragraph which appears to even remotely relate to a configuration file which includes information relating to the properties of textual data. Further, other than merely citing the above passage, the Examiner has provided absolutely no explanation in the Final Office Action as to how the Examiner believes the Hutsch reference discloses a “text properties configuration file.” *See generally*, Final Office Action, pages 6-7, 9.

In the Advisory Action, the Examiner further cited paragraph 185 of the Hutsch reference in support of his position. Specifically, the Examiner stated:

The examiner respectfully disagrees with Applicants’ assertions. In paragraph [0029], the actions performed by the XML decision tree include user device properties and resource properties, which determine the fonts, display parameters on user device ([0185], configuring text properties to be associated with display capabilities of particular devices).

Advisory Action, page 3.

In other words, the Examiner appears to assert that paragraph 185 of the Hutsch reference explicitly sets forth that the XML decision tree discussed in paragraph 29 includes information pertaining to *text properties* (e.g., fonts). After carefully reviewing this cited passage, Appellants respectfully disagree.

The additional passage relied upon by the Examiner states the following:

In response to the request from the lightweight remote visualization component on user device 102i, the infrastructure generation service first issues a request to the client factory on user device 102i to create a remote frame window, and then this service creates a corresponding server window object on web server 320. *The server window object queries the remote frame window on user device 102i to determine the fonts, display parameters, etc. on user device 102i.* Alternatively, the server window object can obtain identifier information from user device 102i and then use this identifier information to access a database that includes the display capabilities of device 102i via configuration server 336.

Hutsch, paragraph 185. (Emphasis added).

After carefully reviewing the above passage, Appellants are unable to identify any teaching with regard to *the configuration file originating from a backend store* as being a text properties configuration file. Indeed, the cited passage does not even mention the XML decision tree discussed above and alluded to by the Examiner. To the contrary, to the extent that this passage appears to even discuss text properties, the passage indicates that the web server 320, by way of a window object, may *query a client device* (e.g., user device 102i) for information pertaining to fonts and display parameters. As the Board will appreciate, even if such information is stored on the client device in the form of a configuration file, the information would have originated *from the client device* and not from a backend store, as recited by claims 2, 9, and 16.

For at least the above reasons, Appellants respectfully maintain that the Hutsch reference fails to teach or suggest all elements of claims 2, 9, and 16. Accordingly,

Appellants request that the Board overturn the Examiner's rejection of claims 2, 9, and 16 under 35 U.S.C. § 102(b) and instruct the Examiner to allow these claims.

Features Omitted from Dependent Claims 3, 10, and 17

Claims 3, 10, and 17 depend from independent claims 1, 8, and 15, respectively, and each generally recite that configuration information may be stored by the configurator as a singleton object. A singleton object is described in the present application as being:

...an object that exists in memory such that *only one of that type of object exists at any time in memory*. Once created, a singleton object is not destroyed after use, like most objects, but is kept in memory until accessed again.

Application, paragraph 37.

For instance, once the appropriate properties or configuration file or files are located by a servlet corresponding to a particular configurator class, the configuration information may be stored as a singleton object, also referred to as a "mapper." *See id.* at paragraph 38. As the Board will appreciate, singleton objects are generally static, so that the configuration information remains consistent for subsequent similar requests. Indeed, the present application notes that configuration information, such as error code tables or logging information, are "*static and do not change* after the web application has started." *Id.* at paragraph 34. That is, if configuration information obtained in response to a particular client request and stored in a singleton object were to be modified, then the configuration information may be inaccurate or invalid when accessed again by the

configurator when processing a subsequent similar client request. Accordingly, Appellants respectfully submit that one skilled in the art will readily acknowledge that a singleton object is *static* once created in memory.

In the rejecting claims 3, 10, and 17 in the Final Office Action, the Examiner appears to have cited the DOM tree 1570 of the configuration service 336 as being analogous to the recited “singleton object.” *See* Final Office Action, pages 6, 8-9. The Examiner’s position appears to rely on one particular embodiment of the invention disclosed in the Hutsch reference, which is described as utilizing a single DOM tree 1570 in the cache 1560 of the configuration server 336. *See* Hutsch, paragraph 328. Appellants respectfully disagree with the Examiner’s proposed correlation for the following reasons.

First, in sharp contrast to the above-recited “singleton object,” the use of a single DOM tree 1570, as disclosed by the Hutsch reference, is for the purpose of *tracking configuration data modifications* by various proxies. *See id.* Specifically, the configuration proxies 1510 are described as functioning to cache the configuration data in the DOM tree 1570 into separate DOM tree structures respective to each configuration proxy 1510. *See id.* at paragraph 327. Thereafter, *modifications* to the configuration data, depending on each different client request, are initially carried out on the DOM tree stored in a corresponding configuration proxy 1510. *See id.* The proxy 1510 then reports the configuration *changes* to the configuration server 336, which in turn *implements the modifications to the DOM tree 1570*. Indeed, the Hutsch reference makes it clear that the

single DOM tree 1570 in the embodiment disclosed in paragraphs 327 and 328 is anything but static.

Second, it appears that each configuration proxy 1510 caches its' own respective "copy" of the original DOM tree 1570 created by the configuration server 336. As discussed above, the separate DOM trees *cached by each of the configuration proxies* 1510 serve to receive configuration data *modifications*, which are later merged with the DOM tree 1570 in the configuration server 336 ("master" DOM tree). *See id.* These teachings would appear to suggest that each *separate* DOM tree must initially be a *copy* of the master DOM tree 1570. That is, each separately cached DOM tree created by each configuration proxy 1510 must initially contain the same configuration information as the master DOM tree 1570 cached by the configuration server 336. Therefore, even though the configuration server 336 may include only a single master copy of the DOM tree 1570, the Hutsch reference clearly discloses that additional copies of the master DOM tree, which are created and cached by each configuration proxy 1510, also exist *concurrently* in memory. As such, Appellants submit that the Examiner's assertion that the master DOM tree 1570 constitutes a "singleton object" clearly contradicts the plain teachings of the cited reference.

In the Advisory Action, the Examiner, in response to Appellants' arguments, further relied on paragraph 239 of the Hutsch reference as allegedly disclosing the recited "singleton object." Specifically, the Examiner stated:

As acknowledged by the Applicants “a singleton object is kept in memory unit accessed again” (i.e., cached in memory).

Paragraph [0239] explicitly teaches device properties have been cached "for easy access"; and paragraphs [0327]-[0329] further teach caching configuration data in a DOM tree 1570 in cache 1560 as a cached object (a singleton object).

Advisory Action, page 3.

First, Appellants note that the Examiner’s proposed definition of “singleton object” conveniently omits certain key features of the term. As discussed above, a singleton object, as clearly defined by the present application, is *not only* an object that is cached in memory, but one that is cached such that *only one type of that object exists at any time in memory*.

With the above distinction in mind, Appellants note that the additional passage relied upon by the Examiner in the Advisory Action merely states:

Device properties are the properties of client device 102i. Device properties refer to such things as display size on the device. These details are stored in configuration service 336 of network portal system 100 and in one embodiment are cached in profiling service configuration file 802 *for easy access*, by the components of profiling service 801, as required.

Hutsch, paragraph 239. (Emphasis added).

As best understood by Appellants, the cited passage merely discloses that profiling service configuration files may be cached for access by components of the profiling service 801. However, there does not appear to be any indication, either in the cited

passage or elsewhere, which supports that the profiling service configuration file is a singleton object that is cached such that only *one type of that object exists in memory at a given time*.

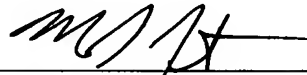
For at least the above reasons, Appellants respectfully maintain that the Hutsch reference fails to teach or suggest all elements of claims 3, 10, and 17. Accordingly, Appellants request that the Board overturn the Examiner's rejection of claims 3, 10, and 17 under 35 U.S.C. § 102(b) and instruct the Examiner to allow these claims.

Conclusion

Appellants respectfully submit that all pending claims are in condition for allowance. However, if the Board wishes to resolve any issues by way of a telephone conference, the Board is kindly invited to contact the undersigned attorney at the telephone number indicated below.

Respectfully submitted,

Date: October 6, 2008



Michael G. Fletcher
Reg. No. 32,777
FLETCHER YODER
(281) 970-4545

CORRESPONDENCE ADDRESS:

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Department, M/S 35
P.O. Box 272400
Fort Collins, Colorado 80527-2400

8. **APPENDIX OF CLAIMS ON APPEAL**

Listing of Claims:

1. A system for creating web applications, the system comprising:
a controller generator that is adapted to provide a web application with a
controller that receives a request for data from a user and responds to the
request by sending information to the user; and
a configurator generator that is adapted to provide a configurator that loads
configuration information for use by the controller from a cached
configuration file that originated from a backend data store, and wherein
the configurator stores the configuration information for subsequent
access.
2. The system set forth in claim 1, wherein the configuration file is a text
properties configuration file.
3. The system set forth in claim 1, wherein the configurator is adapted to
store the configuration information as a singleton object.
4. The system set forth in claim 1, wherein the configuration information
comprises error handling information.

5. The system set forth in claim 1, wherein the configuration information comprises log processing information.
6. The system set forth in claim 1, wherein the configuration information comprises data that is specific to each of a plurality of portals.
7. The system set forth in claim 1, wherein the configurator is adapted to read the configuration information upon initialization of the controller.
8. A method of creating web applications, the method comprising:
creating, with a processor-based device, a controller that receives a request for data from a user and responds to the request by sending information to the user; and
providing a configurator that loads configuration information for use by the controller from a cached configuration file that originated from a backend data store, and wherein the configurator stores the configuration information for subsequent access.
9. The method set forth in claim 8, comprising defining the configuration file to be a text properties configuration file.
10. The method set forth in claim 8, comprising adapting the configurator to store the configuration information as a singleton object.

11. The method set forth in claim 8, comprising defining the configuration information to comprise error handling information.

12. The method set forth in claim 8, comprising defining the configuration information to comprise log processing information.

13. The method set forth in claim 8, comprising defining the configuration information to comprise data that is specific to each of a plurality of portals.

14. The method set forth in claim 8, comprising adapting the configurator to read the configuration information upon initialization of the controller.

15. A system for creating web applications, the system comprising:
means for creating a controller that is adapted to receive a request for data from a user and respond to the request; and
means for creating a configurator that loads configuration information for use by the controller from a cached configuration file that originated from a backend data store, and wherein the configurator stores the configuration information for subsequent access.

16. The system set forth in claim 15, wherein the configuration file is a text properties configuration file.

17. The system set forth in claim 15, wherein the configurator is adapted to store the configuration information as a singleton object.

18. The system set forth in claim 15, wherein the configuration information comprises error handling information.

19. The system set forth in claim 15, wherein the configuration information comprises log processing information.

20. The system set forth in claim 15, wherein the configuration information comprises data that is specific to each of a plurality of portals.

21. The system set forth in claim 15, wherein the configurator is adapted to read the configuration information upon initialization of the controller.

22. A machine readable medium, comprising:

a controller generator stored on the machine readable medium, the controller

generator being adapted to provide a web application with a controller that

receives a request for data from a user and responds to the request by

sending information to the user; and

a configurator generator stored on the machine readable medium, the configurator generator being adapted to provide a configurator that loads configuration information for use by the controller from a cached configuration file that originated from a backend data store, and wherein the configurator stores the configuration information for subsequent access.

23. The machine readable medium set forth in claim 22, wherein the configurator generator is adapted to produce a configurator that stores the configuration information as a singleton object.

24. The machine readable medium set forth in claim 22, wherein the configurator generator is adapted to produce a configurator that reads the configuration information upon initialization of the controller.

9. **EVIDENCE APPENDIX**

Exhibit A:

Definition 1d of “system,” *Webster’s Ninth New Collegiate Dictionary*, page 1199 (1989).

10. **RELATED PROCEEDINGS APPENDIX**

None.

EXHIBIT A



WEBSTER'S
Ninth New
Collegiate
Dictionary



A GENUINE MERRIAM-WEBSTER

The name *Webster* alone is no guarantee of excellence. It is used by a number of publishers and may serve mainly to mislead an unwary buyer.

A *Merriam-Webster*® is the registered trademark you should look for when you consider the purchase of dictionaries or other fine reference books. It carries the reputation of a company that has been publishing since 1831 and is your assurance of quality and authority.

Copyright © 1989 by Merriam-Webster Inc.

Philippines Copyright 1989 by Merriam-Webster Inc.

Library of Congress Cataloging in Publication Data
Main entry under title:

Webster's ninth new collegiate dictionary.

Includes index.

I. English language—Dictionaries. I. Merriam-Webster Inc.

PE1628.W5638 1989 423 88-8335

ISBN 0-87779-508-8

ISBN 0-87779-509-6 (indexed)

ISBN 0-87779-510-X (deluxe)

Abbrev

Webster's Ninth New Collegiate Dictionary principal copyright 1983

COLLEGIATE trademark Reg. U.S. Pat. Off.

All rights reserved. No part of this book covered by the copyrights hereon may be reproduced or copied in any form or by any means—graphic, electronic, or mechanical, including photocopying, taping, or information storage and retrieval systems—without written permission of the publisher.

Made in the United States of America

32333435RMcn89

